

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/259496289>

History and comparative study of modern game engines

Article · January 2012

CITATIONS

31

READS

11,357

3 authors:



[Partha Sarathi Paul](#)

7 PUBLICATIONS 41 CITATIONS

[SEE PROFILE](#)



[Surajit Goon](#)

BCDA College of Pharmacy and Technology

15 PUBLICATIONS 52 CITATIONS

[SEE PROFILE](#)



[Abhishek Bhattacharya](#)

Institute of Engineering & Management

55 PUBLICATIONS 455 CITATIONS

[SEE PROFILE](#)

HISTORY AND COMPARATIVE STUDY OF MODERN GAME ENGINES

Partha Sarathi Paul¹, Surajit Goon¹, Abhishek Bhattacharya²

¹Birla Institute of Technology, Kolkata Campus, Kol-107, WB, India

²Institute of Engineering & Management, Salt Lake, Kol-91, WB, India

[Received-29/03/2012, Accepted-21/05/2012]

ABSTRACT:

In this paper we try to explain what actually a game engine means and what are the major components of it. It also describes some special types of game engines with their features. This paper also presents a brief history of game engines evolution process and discusses about some modern game engines specifications with comparison.

Keywords: Game Engines, AI, Middleware, Physics Engine, 3D Engine, Rendering.

DEFINITION AND COMPONENTS:

Generally, the concept of game engine^[1] is very simple to understand. It is a platform of doing common game related tasks like rendering, physics related computation and input, so that developers (artists, designers, scripter and other programmers) can focus on the details that make their game unique. Engines are actually a collection of reusable components that can be manipulated to bring a game to reality. There are some clear differences between a game and a game engine^[2]. Rendering, loading, animation, collision detection between objects, physics, inputs, GUI and AI are the primary different components of an engine. On the other hand

the content of a game, its specific characters and terrains, the reason behind collision, real world object behaviors etc. are the components that make the actual game.

Game engines are like middleware solutions^[3]. It provides platform abstraction, allowing the same game to be played on various platforms including game console and personal computers with only few changes made to the source code. Often game engines are designed with more specialized game middleware^[1] components like Havok^[4] for physics, Miles Sound System^[5] for sounds or Bink^[6] for videos. Physics engines are very popular and significant middleware in game development because they give us the more realistic game play environment. It is

specially created to calculate the physics related numerical computations while using with the overall game engine. These game engines create the simulation of real world physics in the digital world by handling the basic physics system including collision detection, object deformation and dynamics. A recent survey by gamesutra.com found that generally 55% of the people are using a middleware game engine on their project^[7].

SOME SPECIAL TYPE ENGINES:

1. Game engines for Massively Multiplayer Online Games (MMOs,MMOGs) is for more complex application than a single player game. Every individual game can be used to implements a MMO middleware. Some MMO middleware software packages already include a game engine while others provide networking facilities only. Some popular MMO middlewares are –Abyssal engine, Hero engine,Q, Vison engine etc. **2.**

Another subset of game engines are 3D first person shooter(FPS) game engines. It was enhanced through a series of implementation at various levels and generations, like “Jurassic Park:Trespasser”(1998) introduced physics to the FPS games for the first time. Next Red Faction (2001) featured destructive walls and grounds. Battlezone(1998) and Battlezone II: combat commander(1990) added vehicle based combat to FPS games and combination of all the above are implemented later in Tribes 2, Battlefield 1942, Halo: combat evolved and Unreal Tournament 2004. **3.** Visual Novel Engines: these type of engines are not at all resource hungry because they are less graphic-intensive in nature. Examples of these are Visual Novelty, KiriKiri, NScripter etc.

EVOLUTION OF GAME ENGINES:

In early 1989 one sci-fi game engine named Ultima Underworld was developed. That engine had the same name as the game itself.

But after releasing of Space Rogue game, Origin System had undertaken the Ultima Underworld engine and developed the algorithm for texture mapping which can be applied to floors, ceilings, walls etc. The maximum system requirement for this was 386 based PC. In 1993, ID software developed Doom engine , which is not a 3D engine at all, but had capacity to represent objects, characters and whole level map by 2D sprite representation. Rendering was very fast and it needs 386 based PC with standard VGA support to run. Though it was a 2D engine, the illusion created by the developer made it a 3D title. NovaLogic’s prprietary engine Voxel engine (1992) was the basic engines for all Comanche games. Voxel had their own way to represent volumetric objects as three dimensional bitmaps. Before that all engines applied vector graphics, which was little bit slower in speed and less detailing than 3D bitmap presentation. Blade Runner, Command & Conquer are the noteworthy games developed by Voxel engine. In late 1993, another game called Duke Nukem 3D was released into the market ,which was developed with the help of Build engine. Like Doom engine it also created 3D effects in 2D interface. It simply varied the sectors with different heights to achieve the illusions. By applying special tags to various spot within a particular sector, developers could make it so that whenever a player move over to that particular spots it switches over to a different sector giving the illusion of changing the levels or environment at run time. XnGine(1995) was the first ever 3D engine which is developed in DOS base. It would later make use of high resolution graphics and be compatible with 3dfx video cards. Quake engines(1996) was the first truly 3D game engine by Id software. It had an unique processing capacity to render maps by purging certain areas from processing that the

player wouldn't be able to see. It actually took advantages of Z-buffering, which simply is a method for determining which parts of the maps are visible to the player and only rendering those sections. Renderware(1996) was the most popular engine for multiplatform games. It supports PlayStation 2, Wii, GameCube, Xbox, Xbox 360, PlayStation 3 and PSP platforms. Quake II/ id Tech 2 engine(1997) supports native OpenGL, colored lightening effects and C language support. The moddability was increased because of its DLL support. GoldSRC(1998) pushed PC games to a new era. It supports both OpenGL and Direct3D. Some successful game examples which are based on OpenGL are Half-Life, Day of Defeat and Counter Strike etc. One of the most popular game engine is Unreal engine(1998), which gave birth to Unreal Tournament game. It integrates its own scripting language called UnrealScript and map editor named UnrealEd. A modified version of Quake II engine was Quake III designed in 1999 which supports 32-bit color, shaders and advance networking. John Slagel, *Red Faction's* lead programmer, developed the Geodmod engine(2001), which stands for Geometry Modification. As *Red Faction's* lead designer Alan Lawrence explained to Gamespot, "When a rocket hits a wall, we take this shape and basically subtract that shape from the world. So we boolean with that 'bit' -- we call them GeoMod bits -- and that takes a chunk out of the world."^[8]. In 2001 Torque engine was developed to modify the FPS Tribes 2 game. It had on the fly rendering option with less polygon counts and also had a world map editor integrated within it. Serious engine(2001) was designed to allow large spaces and large numbers of on-screen characters at any given time and give birth to popular Serious Sam series. Later in Doom 3

(2004) most light surfaces were also done in real-time, allowing more realistic shadows, but at the expense of being able to render soft shadows. To get around this, projected lights could be used to create the illusion of soft shadows. Half Life 2 creator Source engine (2005) including advanced shader technologies, dynamic lighting and shadows, physics, several effects such as realistic looking reflective water surfaces and real-time motion blur, and much more. CryEngine's(2004) used pixel shaders for realistic water in Far Cry. Developed by Crytek company it produced the game named Crysis, a shader heavy DirectX 10 game. Rockstar Advanced Game Engine(RAGE)(2006) combines a rendering framework, physics engine, audio engine, network libraries, animation engine and scripting language in one package. Grand Theft Auto IV was the finest product of that particular engine. In 2007 Unreal engine and 2008 Frostbite engine took the advancement of game engines to the next level. Battlefield: Bad Company, a game which increased it and up to 92 percent of the environment can be destroyed with unbelievable blasting effects was created by using Frostbite engine^[7]. Cry engine 3 by Crytek was launched in 14 Oct'2009 which is again a cross platform engine supports C/C++. As for the PC platform, the engine is said to support development in DirectX 9, 10, and 11. As of June 1, 2009, it was announced that Crysis 2 would be developed by Crytek on their brand new engine.

COMPARISON:

Comparison between various game engine is not a simple task because of their various genre, types, multimedia support, middleware support, language and platform dependencies, rendering techniques and many other sub features^[9]. In this paper we tried to give some idea about the strengths and weakness of

some well-known game engines. We took six popular game engines to compare^[10] with each another and they are Cry Engine, Hero Engine, Source Engine, Unity Engine, Unreal Engine 3 and Vision Engine. Table-1 describes the performances in terms of platform dependencies, Interface supports, API supports, inbuilt physics engine supports with path finding capabilities and Table-2 indicates the World and GUI editor supports, rendering techniques, and inbuilt MMO supports of those selected engines.

Table-1:

Game Engines	Platform	Language Support	API	Physics Engine	Path finding
CryEngine	Win,X360, PS3	C++, Visual Script	DX9, 10,11	Custom	Yes
Hero Engine	Win	C++, Hero Script	DX 9	Physx	Yes
Source Engine	Win, Mac, X360, PS3	C++	DX9, Open GL	Custom	Yes
Unity	Win, Mac, Android, X360, PS3, Wii	C#, Java Script	DX9, Open GL	Physx	No
Unreal Engine 3	Win, X360, PS3	C++, Unreal Script	DX9, 10, 11	Physx	Yes
Vision Engine	Win, X360, PS3, Wii	C++	DX9, 10, 11	Physx, Bullet	Yes

Table-2

Engines	World Editor	GUI Editor	Deferred renderer	Forward renderer	Built-in MMO
Cry Engine	Yes	No	Yes	No	No
Hero Engine	Yes	No	No	Yes	Yes
Source Engine	Yes	No	No	Yes	No
Unity	Yes	No	Yes	Yes	No
Unreal Engine 3	Yes	Yes	Yes	No	No
Vision Engine	Yes	No	Yes	Yes	No

A survey was done with the help of 100 executives from different game industries^[7] who had given their opinions on various game engines presently available in the market. It indicates that the most popular

platform is X360 (see Figure-1). The awareness about recent engines is tremendous. 80 to 90 percentage of them are very much aware about the functionalities and developments of the current engines(see Figure-2). But surprisingly they are biased with some particular engines when the ease of practical use is concerned (see Figure-3).

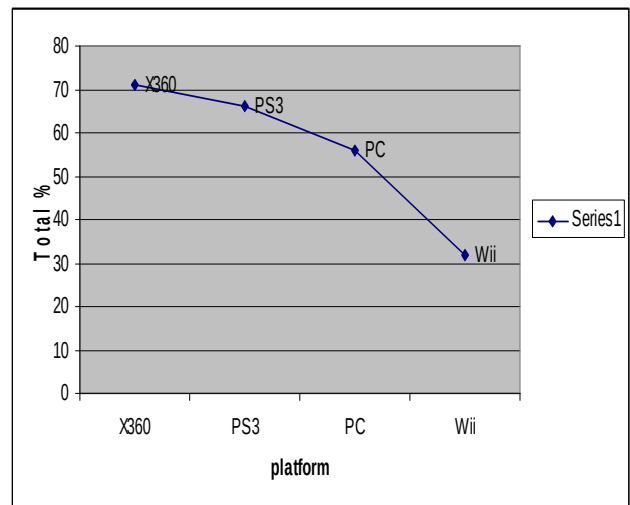


Figure-1. shows the popularity of various platforms of game developments.

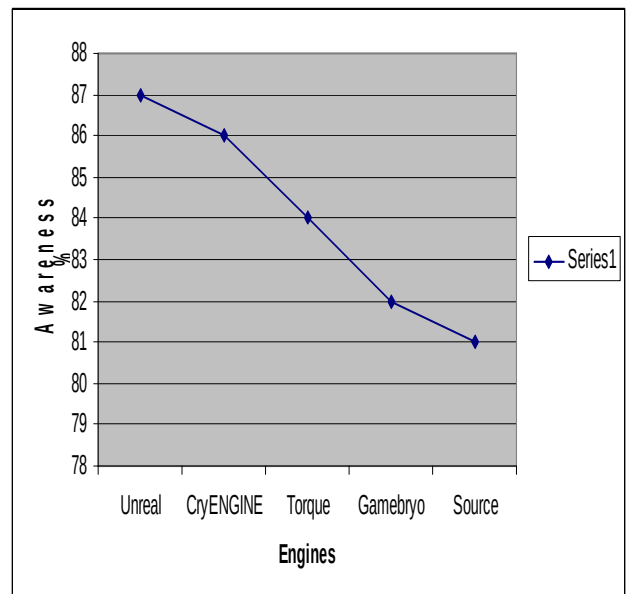


Figure-2. Awareness of popular engines.

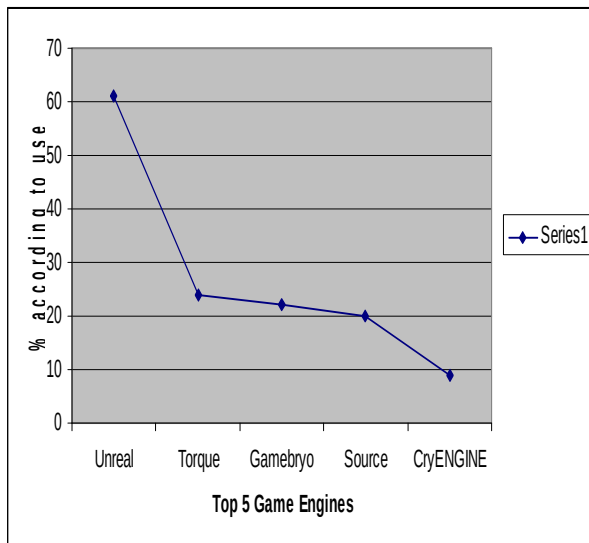


Figure-3. Practical industry acceptance of popular game engines.

CURRENT STATE OF THE ART TRENDS OF GAME ENGINES:

Game engines technology, today, is become more user-friendly then previous few years and it is continuously changing with time. Recently in spite of the traditional game development, they are used for some serious business also like training, medical teaching, military simulations etc. They are also now targeted to the multiple platforms like-Android phones, iPhones etc. 3D rich games of recent times are mostly GPU-limited, that means restricted by graphics card's power. Microsoft, the giant of software industry, is also looking forward to this technology. They developed XNA as the SDK for all developers to build their games on X box platforms.

CONCLUSION:

From the beginning of game engines to the recent 3D high performance game engines, the goal of development is remain the same i.e. to give developer a platform to create their unique games into reality. They need

not to write or develop the game from the scratch but just supposed to implement the idea with the help of some engine. In engines the basic core architecture and codes are given, the middlewares are present, a developer only needs to tweak them as their own requirement. The evolution of game engines is now moving towards more realistic and technically rich games in various fields like physics, sounds, and rendering etc.

REFERENCES:

1. http://en.wikipedia.org/wiki/Game_engine.
2. Dissecting the Video Game Engine and a Brief History by Julie Wilmore Retrieved From-
juliewilmore.files.wordpress.com/2010/07/tc339finalpaper.pdf
3. <http://www.iwaysoftware.de/middleware.html>
4. <http://www.havok.com/products/physics>
5. <http://www.radgametools.com/miles.htm>
6. <http://www.radgametools.com/bnkmmain.htm>
7. DeLaura, Mark. March 16, 2009. The Engine Survey: Technology Results. Retrieved From
http://www.gamasutra.com/blogs/MarkDeLaura/20090316/903/The_Engine_Survey_Technology_Rresults.php.
8. Lilly, Pauly. July 17, 2009. Doom to Dunia: A Visual History of Game Engines.Maximum PC. Retrieved from --
http://www.maximumpc.com/article/features/3d_game_engines?page=0%2C0.
9. Simpson, Jake. July 11, 2002. Game Engine Anatomy 101. Retrieved from -
<http://juliewilmore.files.wordpress.com/2010/07/tc339finalpaper.pdf>
10. "Game Engine Comparison"-Retrieved from
<http://www.esenthel.com/?id=compare>